STATEMENT OF WORK

Title: Technical Support for Development of Multimedia Modeling Systems and Integration with SuperMUSE V1, D4EM, and FRAMES V2 Infrastructure Software Support Systems

Contractor and Contract No.: EP-W-08-019

Work Assignment No.: 4-3

Estimated Level of Effort: 6,921 hours

EPA Key Personnel:

Work Assignment Manager (WAM):

Justin Babendreier
USEPA National Exposure Research Laboratory/ORD
Ecosystems Research Division
960 College Station Road
Athens, GA 30605-2700
Phone: 706-355-8344
Email: babendreier.justin@epa.gov

Project Officer:

Ryan Daniels
1200 Pennsylvania Ave. NW (3803R)
Washington, D.C. 20460
Phone (202) 564-6476
E-mail: daniels.ryan@epa.gov

Research Programs for CO2 Sequestration and Sensitivity and Uncertainty Analyses

The primary aim of this work is to develop effective, integrated, place-based Source-to-Outcome modeling strategies for quality assured exposure and risk assessment of CO2 sequestration activities on the landscape (e.g., assessing impacts to water quantity and quality from CO2 injection wells). A key underlying context in this work is development of capacity to assess single stressor outcomes (e.g., injected CO2) in the context of systems concurrently impacted by multiple-stressors (e.g., those also dealing with toxicants, nutrients, sedimentation, etc). This work will be aided by the contractor through execution of the following software development activities:

1) Support, development, and assimilation of select components of 3MRA1.x (i.e. models, data, processors, tools) into FRAMESv2, constituting 3MRAv2,
2) Support, development, and assimilation of select components of other integrated modeling systems (e.g., models, data, processors, tools in iemWatersheds, etc.) into FRAMESv2, constituting 3MRAv2,
3) Assimilate, apply and test various models and modeling components for CO2 sequestration evaluation, along with other related watershed-scale modeling systems (e.g., pressure front models, GIS data tools, etc) within FRAMESv2,
4) Integrate CO2 sequestration "source term" models with science and data components of 3MRAv2 and other FRAMESv2 domains (e.g., EARTH, iemWatersheds, MIRA, etc) to support exposure and risk assessment capacity.
5) Integrate this overall extended modeling system (e.g. FRAMESv2 CO2 Domain) with various model evaluation tools and experimental simulation strategies (e.g., UA/SA/PE tools in F2, parallel computing afforded via SuperMUSE, etc).
6) Investigate design strategies to extend SuperMUSEv1 "tasking" concepts for direct support of 64 bit parallel processing on single desktop platforms.
7) Create a software-based preprocessor strategy and toolset for D4EM and iemTechnologies modeling systems (i.e. OpenTERRAworks software) that will more easily facilitate cumulative impact assessments in projects that implement large scale earthworks design that materially alter actual topography and resultant hydrology of systems.

EPA is pursuing a three-tiered strategy for development of overall CO2 sequestration evaluation tools that range from simpler web-based tools to single model desktop tools to the more integrated "framed" modeling system schemes described above. It is the intention of this work to build an overall complementary approach that facilitates users of these tool schemes across associated levels of capability and complexity in software formulation and use.

EPA/ORD/NERL/ERD's research program for investigating sensitivity and uncertainty analyses for various environmental models currently utilizes a series of 400 PCs linked together in a local area network. This bank of PCs, a functional equivalent to a supercomputer, allows for computationally intensive modeling experiments to be conducted. The methodology focuses on computing many simulations of a single model or modeling system application. The cluster is referred to as SuperMUSE – Supercomputer for Model Uncertainty and Sensitivity Evaluation.

The PC cluster and associated management software currently support 32-bit Windows-based operating system environs, and are capable of supporting Linux-based operating systems. To fully utilize this network of PCs, a variety of software tools have been developed using a standard database structure based on contemporary open-

source MySQL. Many of the tools are model-independent, where example model dependent prototypes have also initially been developed for simulation of Version 1.x of the FRAMES 3MRA modeling technology.

In summary, this statement of work covers development, assimilation, maintenance and enhancement of CO2 sequestration models, data, and tools, and FRAMESv2, SuperMUSE 1.0 and 3MRA 1.x/2.x models, data, and tools. This includes work on models and tools associated with those systems, software development support for additional environmental models and data to be assimilated as needed, and tools for uncertainty and sensitivity assessment.

OpenTERRAworks efforts as a preprocessor set for all these tools to be developed under here represent an initial phase for studying air emissions of PM and toxics (at mesoscale to microscale contexts), as well as handling overall changes to hydrologic and hydraulic regimes -- for example, as associated with actual mountaintop removal and construction activities that may occur together with CO2 sequestration and hydrofracking activities. Work under this scope would be inclusive also of meteorology modeling needed to drive integrated modeling work (e.g., AERMOD and WRF) that may be part of an overall approach to support NEPA-based cumulative impact perspectives.

Background

The Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES) - Multimedia, Multipathway, Multireceptor Risk Analysis (3MRA) software system is an integrated multimedia modeling system for assessing exposure and risks from the release of hazardous materials placed into a variety of land-based waste management units. The FRAMES 3MRA Version 1.0 (FRAMES 3MRA 1.0) software system was constructed to perform risk analyses for the U.S. Environmental Protection Agency (EPA) Office of Solid Waste to help establish constituent-specific “exit” (e.g., safe disposal) levels for low risk solid wastes. In the design of FRAMES 3MRA, the component-based approach provides for 1) standardized tools and techniques that are typically used in the assessment process, and 2) capabilities for new functionality to be added.

The FRAMES 3MRA 1.0 was originally designed to run on a single PC computer system. It was found that parallel execution across a number of machines would be valuable, helping to expedite simulation experiments needed for large, national-scale studies and various uncertainty and sensitivity analysis studies. The FRAMES 3MRA 1.x version of the software was designed and built to allow for, among other capabilities, parallel execution of the FRAMES 3MRA 1.0 modeling system across multiple machines. FRAMES 3MRA Version 2.0 software components, covered under this scope as well, represents a further, significantly enhanced software technology that replaces the system user interface with a more generic user interface concept.

To successfully control and implement the FRAMES 3MRA 1.x system so multiple (e.g., millions) runs can be simultaneously executed and tracked on the 400+ machines, a number of software tools have and are being developed to help manage the operation of the system, as well track files, warnings, and errors. Because the vocabulary can be daunting at times, a number of key components are defined as follows:

- **Aggregated Exit Level Processor II Visualization (AggELP2Vis)—**The AggELP2Vis is a program that performs many of the same operations as the AggELP2MySQL, but instead renders a hypertext markup language (HTML) document that shows all the scenarios in a single context. The original ELP2/RVP allows a user to see one chart at a time, whereas the AggELP2Vis allows the user to see all scenarios and impacts on populations, cohorts, distances, exposures, and receptors that are not specifically protected. A GNUPlot is used to generate the charts.

- **Aggregated Exit Level Processor I for MySQL (AggELP1MySQL)—**The AggELP1MySQL is a program logically identical to the original ELP1 with the simple change that the information is stored in a My Structured Query Language (MySQL) database instead of MS-Access. The resulting file is used as input to the AggELP2MySQL and the AggClientCollect.

- **Aggregated Exit Level Processor II for MySQL (AggELP2MySQL)—**The AggELP2MySQL is a program logically identical to the original ELP2/RVP that reads its inputs from the MySQL database. The results are tables stored in MySQL that are equivalent to the original Protective Summary Output Files. Setting the scenarios in this tool facilitates the AggELP2Vis in displaying all the scenarios simultaneously.

- **Andres Iterated Fractional Factorial Design Dynamic Link Library (AIFFDDLL)—**The AIFFDDLL is the Enhanced Computational Optimization Sensitivity Uncertainty(ECOSU) compliant implementation of a grouping and input changing strategy that seeks to determine which variables in a large number of variables change the output the most. It is a screening method for finding the most sensitive parameters. The AIFFDDLL is delivered as a set of subroutines and functions that are part of a dynamic library.

- **Batch Tasker—**This is a Model Tasker similar to the Command Tasker but without the restriction of executing commands on specific machines. The Batch Tasker consumes a text file where each line is a command. Each command is invoked in order on the next available machine.

- **Central Processing Unit Allocator (CPU Allocator or CPUa)—**The CPUa is responsible for making sure available machines are assigned to a Model Tasker running on some machine in the cluster. Every machine is a slave to a specific

CPUa, as there can be multiple CPUa's. The Model Tasker, CPUa, and Tasker Client continuously communicate with each other.

- **Client Collector for Aggregated Exit Level Processor I (CCAggELP1)—**The CCAggELP1 is an application that reads two instances of the output of the AggELP1and merges them into a single instance. This is used to collect the output of ELP1 in pairs across the cluster of machines. The CCAggELP1 is intended to collect simulation results from another single machine. A collection across a large number of machines can then be done by simply using CCAggELP1 to collect the results in pairs and then collect those results into pairs again, and so on until all the results are accumulated on a single machine. The client collect tool takes the results that are produced on each individual client and compiles them into a single database.

- **Command Tasker—**The Command Tasker is a specialized Tasker that is essentially a server-end batch file manager. The Command Tasker executes commands on specific machines in a specific order. It provides machine-specific commands, based on a set of prerequisite tasks and takes a series of commands, but confirms that specified previous commands have already been executed, thereby accounting for dependent commands (e.g., delete files in a certain order). Similar to the capabilities of the Update Client tool, and actually representing a Model Tasker, this tool delivers binary tree task dependencies in a collection of common aggregated data/files, or reversibly, in distribution of common data/files. The Command Tasker acts as a Model Tasker in managing activities across the cluster, allowing the user to issue commands to clients (e.g., DOS commands for Windows or shell scripts for Linux) that are executed by the Tasker Client. Extensively generic in form, it is currently used for conducting log-scale database collections for 3MRA experiments and for more quickly executing file-management tasks that take individual PCs substantial time to complete.

- **Delegating Dynamic Link Library (DDLL)—**This library is a single entry point for any ECOSU compliment DLL to provide sampling algorithms. For example, if Monte Carlo is chosen as a sampling approach, the DLL is responsible for redirecting all calls to sampling algorithms and all results to the actual Monte Carlo functionality.

- **Enhanced System User Interface (ESUI)—**The ESUI provides the user with an enhanced ability to pick and choose specific input combinations of chemical name, site ID, realization, and concentration of waste (Cw), so only that specific run or set of runs are executed and where one does not have to look through a large number of simulation sets to get to the specific run. All information is stored in the 3MRA header file [hd.ssf].

- **Enhanced 3MRA Chemical Properties Processor—**This is a logically identical chemical property processor that reads its input data from the cp.ssf file instead of reading the ASCII data file originally stored in the CPData directory. The site definition processor (SDP) will read the cp parameters as any other component; it

will then call the enhanced CPPDLL. The CPPDLL is responsible for populating all the original values in the cp.ssf datafile from the data provided in the cp.ssf from the SDP.

- **Enhanced 3MRA SUI Deterministic Switch**—This is an addition to the Enhanced System User Interface (ESUI) that allows the user to choose the sampling technique and whether full sampling is accomplished or just a deterministic run. Under FY05 development, the ESUI will be able to run the DSP and allow the user to change the sampling algorithms as well.

- **Enhanced 3MRA SDP Deterministic Switch**—This switch is an addition to the SDP that allows the use of central tendency instead of actually sampling the value from the distribution. The changing value of the sampling technique is passed onto the Delegating Dynamic Link Library via this SDP enhancement.

- **FRAMES-2.0**—The Framework for Risk Analysis in Multimedia Environmental Systems- Version 2.0 (FRAMES-2.0) is a system that allows legacy disparate models and databases to communicate in a plug and play atmosphere. It combines many of the best features of FRAMES version 1 (e.g., Framework User Interface) and FRAMES 3MRA 1.0 (e.g., Application Programming Interface).

- **FRAMES 3MRA**—The Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES)-Multimedia, Multipathway, Multireceptor Risk Analysis (3MRA) software system is an integrated multimedia modeling system for assessing exposure and risks from the release of hazardous materials placed into a variety of land-based waste management units.

- **FRAMES 3MRA 1.0**—The FRAMES 3MRA Version 1.0 software system was constructed to perform risk analyses for the EPA Office of Solid Waste to help establish constituent-specific "exit" (e.g., safe disposal) levels for low risk solid wastes. In the design of FRAMES 3MRA, the component-based approach provides for 1) standardized tools and techniques that are typically used in the assessment process, and 2) capabilities for new functionality to be added. The FRAMES 3MRA 1.0 was originally designed to run on a single PC computer system.

- **FRAMES 3MRA 1.x**—The FRAMES 3MRA 1.x version of the software was designed and built to allow for, among other capabilities, parallel execution of the 3MRA 1.0 modeling system across multiple machines. It was found that parallel execution across a number of machines would be valuable, helping to expedite simulation experiments needed for large, national-scale studies and various uncertainty and sensitivity analysis studies.

- **FRAMES 3MRA 2.0**—The FRAMES 3MRA Version 2.0 represents a further, significantly enhanced version of the FRAMES 3MRA 1.x software technology by replacing the system user interface with a more generic user interface concept.

- **Framework User Interface Tasker (FUITasker)**—The FUITasker modifies module inputs and either wraps the entire file set and sends it to the Tasker Client for further processing or performs the required processing locally. The FUITasker is a single looping capability for Framework for Analysis of Risk in Multimedia Environmental Systems (FRAMES) 2.0 that allows the user to change the value of any single parameter. The looping can be executed on a single computer (called serial mode) or on the cluster (called parallel mode).

- **Latin Hypercube Dynamic Link Library (LHSDLL)**—The LHSDLL is the ECOSU compliant implementation of the Latin Hypercube sampling algorithm. The LHSDLL is delivered as a set of subroutines and functions that are part of a dynamic library.

- **Model Tasker**—The Model Tasker is a type of a component that provides a listing of things to do and resides on some machine in the cluster. There are many examples of this type of component: the Batch Tasker, Command Tasker, SUI Tasker, and FUITasker are actual examples in use. The Model Tasker, CPUa, and Tasker Client continuously communicate with each other.

- **Morris One-at-a-Time Dynamic Link Library (MOATDLL)**—The MOATDLL is the ECOSU compliant implementation of a one at a time input changing strategy associated with Morris. The MOATDLL is delivered as a set of subroutines and functions that are part of a dynamic library.

- **Process Error Program (PEP)**—The PEP is program that is designed to read the errors and warning files produced by FRAMES 3MRA hwirio.dll and store them in a central MySQL database. The PEP is used to keep track of which components in the simulation have succeeded or failed. It provides the user with the ability to capture error and warning messages and store them in the same location as the Site Summary Tool (SST). It works on the assumption that when any component of the system software fails, an error or warning file is produced in the grf directory. The PEP simply copies the Warning or Error file from the grf directory to the MySQL database that is referenced in its command line and, therefore, has no user interface.

- **Refactored Monte Carlo Dynamic Link Library (RMCDLL)**—The RMCDLL is the ECOSU compliant implementation of Monte Carlo sampling. The RMCDLL is delivered as a set of subroutines and functions that are part of a dynamic library.

- **Site Summary Tool User Interface (SSTUI)**—The SSTUI allows the user to pick-and-choose output from a set of 3MRA model input and output files (site simulation file [SSF] and global results file [GRF] files) via the SST. For example, it will you allow you to define how to extract information for a variable for a specific chemical and location but averaged for all times. It allows one to statistically roll-up outputs.

- **Site Visualization**—This is a program that displays a plot of all results that have time as a dependent variable. It starts at the source and ends at human and

ecological exposure. This application uses GNUPlot to generate charts while the application itself creates an HTML document that has the charts organized in a logical manner.

- **Site Summary Tool (SST)**—The SST is a program that allows the user to extract, summarize, and store modeling results in a database. The SST requires the user to create an instruction *.csv script file that describes what information to consume (i.e., extract) from model inputs and outputs for a single FRAMES 3MRA 1.x simulation. The SST extracts information from the SSF and GRF files, given a text file that describes the variable to be extracted and how to summarize those data. The results of the extract and summary are stored in a MySQL database.

- **System User Interface Tasker (SUITasker)**—The SUITasker reads a header file and buffers up compute jobs so no machines are waiting to execute a job. It passes RunAll.bat and then launches Run.bat, which is on all machines.

- **Tasker Client**—The Tasker Client is the workhorse of the parallel software system. It is a generalized batch file execution tool that uses transmission control protocol/Internet protocol (TCP/IP) to get the information about 1) the job it should contribute to and 2) the specific task it needs to perform. The task is communicated in a single Unicode Transformation Format (UTF) string that contains the batch file and a number of additional text files. It runs the actual jobs and is a slave to the CPUa and then to a Model Tasker to complete a computational task. When the Tasker Client has nothing to compute, it goes and finds something to compute from the CPUa. The Model Tasker, CPUa, and Tasker Client continuously communicate with each other.

- **Tasker**—In the parallel software system, a Tasker is any program that generates tasks that need to be performed and registers itself with the CPU Allocator. It is implemented as a TCP/IP server that waits for client machines to be directed to the Tasker by the CPU Allocator.

- **Update Client**—The Update Client 1) prepares the machines for use in the cluster, 2) copies new executables to all machines in the cluster, 3) reads list of computers, and 4) picks computers. Additional features include creating an input file for the command tasker that can collect, distribute, or invoke a command in parallel across the cluster. The Update Client tool facilitates the execution of Operating System (OS) level commands (e.g., DOS/Linux commands, batch/script files) on a large number of machines that comprise a cluster. There are two modes of operation: serial or parallel. The tool can be used, for example, to copy a single file to multiple machines, in serial or in parallel, using a binary tree scheme. In serial mode, it can also be used to perform a variety of file management tasks, such as deletion or alteration of file attributes across a network. The enhanced parallel-mode version can replace an additional set of variables with information from a partner machine.

- **FRAMES V2**— Not specifically listed and described by constituent item here, various tools, processors, models and datasets comprise V2 and form the initial starting basis for work described under this, where many of these components have analogies to those described above for 3MRA V1/V2.

There are several additional tools not delineated above which actually comprise existing CO2 Sequestration models and the entirety of SuperMUSE V1, 3MRA V1/V2, and FRAMES V2 Software Systems.

Tasks:

The following tasks list the specific work required.

Task 1: Workplan Development, QAPP Development and Project Management

The objective of this task is to document a detailed work plan in response to the Work Assignment Statement of Work. The contractor shall document a work plan plan and cost estimate for conducting the assigned work in accordance with the terms of the contract.

Deliverables and Schedule:

1. The contractor shall submit a work plan and cost estimate in accordance with the terms of the contract.

2. The contractor shall submit, in accordance with the terms of the contract, a detailed Quality Assurance Project Plan (category Model Development) describing the project's specific quality assurance project plans to achieving the objectives of the work assignment, and how overall compliance with the QMP for this contract is to be achieved.

3. The contractor shall provide monthly progress reports in accordance with the terms of the contract.

Task 2: Maintenance and Enhancement of CO2 and 3MRAv2 Modeling Domains, SuperMUSE V1, FRAMES V2, and 3MRA V1/V2 Software Systems

The objective of this task is to provide software maintenance and enhancement support for the CO2 and 3MRAv2 modeling domains in FRAMESv2, SuperMUSE 1.0, FRAMES V2 and 3MRA V1/2 software systems.

General Tasking to be Performed

2.1 Software Maintenance Tasking

The contractor shall perform software maintenance tasking which includes:

- Telephone or email communications with the WAM or the WAM's technical support staff.
- Troubleshooting and resolution of bugs identified by EPA, and those bugs that arise out of testing and evaluation performed by the contractor,
- Development and/or revision of spreadsheet-based test plans, and
- Execution of test plans.

2.2 Software Enhancement Tasking

The contractor shall perform software enhancement which includes:

- Telephone or email communications with the WAM or the WAM's technical support staff.
- Modification of existing software to address new requirements specified by EPA,
- Troubleshooting and resolution of bugs identified by EPA during subsequent testing, and those bugs that arise out of testing and evaluation performed by the contractor,
- Development and/or revision of spreadsheet-based test plans, and
- Execution of test plans.

Software documentation and test plans, currently located on USDA's COLAB Development Environment (https://colab.sc.egov.usda.gov/cb/workspace.do; 3MRA FRAMES V2 Project Area) will be the basis for evaluation of existing software requirements and functionality. Additional software requirements associated with component enhancements will be specified by EPA through Technical Direction associated with this statement of work.

Development, modification and/or enhancement of existing documentation (i.e., the formal documents which include sections on descriptions, requirements, design, and specifications) will be the responsibility of EPA or as delegated to the contractor by the WAM.

2.3 Software Development Tasking

The contractor shall perform software development tasking which includes:

- Telephone or email communications with the WAM or the WAM's technical support staff.

- Development of new software to address new sets of requirement specified by EPA,
- Troubleshooting and resolution of bugs identified by EPA during subsequent testing, and those bugs that arise out of testing and evaluation performed by the contractor,
- Development and/or revision of spreadsheet-based test plans, and
- Execution of test plans.

Software documentation and test plans arising out of new development shall be delivered to USDA's COLAB Development Environment (https://colab.sc.egov.usda.gov/cb/workspace.do; 3MRA FRAMES V2 Project Area) and will be the basis for evaluation of new software requirements and functionality. Additional software requirements associated with any subsequent component enhancements will be specified by EPA through Technical Direction associated with this statement of work.

Development, modification and/or enhancement of new documentation (i.e., the formal documents which include sections on descriptions, requirements, design, and specifications) will be the responsibility of EPA or as delegated to the contractor by the WAM.

2.4 Miscellaneous Software Maintenance, Enhancement, and Development Activities

In addition to revision, execution, and documentation of test plans, the contractor shall be responsible for providing brief summary descriptions (using notation and/or file management features of COLAB) on changes to design and specifications sections as may be needed to maintain and/or enhance software (e.g., brief statements indicating information that may need addition/modification, dictionary and/or database table structure definitions that may need addition/modification, etc).

2.5 Travel

Travel is anticipated for some aspects of the work covered under this SOW.

As part of OpenTERRAworks development, a representative core team from the contractor shall attend a 2-day meeting to be hosted by EPA and held in Athens, GA. This will collect Federal Government Agency experts involved in MTM study and design, oversight and regulation. The purpose of this 2-day meeting is to finalize design requirements. Participants may include, but may not be limited to, involvement of staff from ACOE, USGS, EPA, OSM, and FWS. As time and resources allow, one additional 1-day technical requirements implementation design meeting between EPA staff and key software development staff from the contactor will directly follow the 2-day panel meeting (i.e. 3 consecutive days of meetings). The purpose of this last day will be to evaluate final design requirements, and initiate actual software development of the tool.

Technical Direction

In accordance with the terms of the contract, the Agency will provide a written description of each request for work to be completed on specific software components, and the required schedule. These requests will be by Technical Direction and will generally indicate: a) the software component(s) to be tested, de-bugged and/or enhanced, b) initial formulations of any new or modified software requirements, and c) a not-to-exceed number of hours of time, by staff level category (e.g., Senior Software Engineer), that may be expended by the contractor on the given request. EPA will be responsible for posting any initial set of existing bugs to COLAB for software maintenance or enhancement activities. New requirement sets desired by the Agency will be posted to COLAB as a bug, with an indicator that the bug is associated with a new development.

It is anticipated that several components may be associated with a given request, where work on individual components may or may not be directly related. It is also anticipated that more than one technical directive may need to be active at a given time to address new issues that may arise in bringing closure to an existing request.

Because a given bug cannot always be immediately associated with a given component, it is anticipated some components will be specified in the request that ultimately do not need modification.

While fulfilling given Technical Direction, in the event an additional component(s) is identified by the contractor as needing enhancement or modification to achieve the original request, the contractor shall: a) post associated bugs on COLAB; and b) notify the WAM. As determined by the WAM, a new or modified request will be issued to handle associated software enhancements or modifications of the newly identified component.

The contractor may evaluate any existing CO2, SuperMUSE 1.0, FRAMESv2, and 3MRA V1/2 software codes for any component at anytime as needed to execute a given request (including execution of informal software testing by the developer), but shall not post enhanced or modified codes to COLAB, or conduct formal testing of any component, unless that component has been identified in a specific request.

Schedule: Technical direction will be issued in writing or confirmed in writing within five (5) calendar days after verbal issuance. One copy of the technical direction memorandum will be forwarded to the Contracting Officer and the Work Assignment Manager.

Contractor Response to Specific Technical Direction

Prior to initiation of actual bug resolution, enhancement or new development efforts, the contractor will first:

- Review the request,

- As needed review associated codes for components specified in the request, and
- Consult the WAM via telephone to discuss technical content of the request (e.g., to review and modify if necessary newly stated requirements, to discuss current software behaviors needing resolution, and to discuss initial technical approach to be taken to achieve software enhancement or modification).

For each request the contractor shall then execute the required enhancement, modification, development, or testing, and deliver the resulting source code, software, test plans, and summary notations on design and specifications to the Agency via the COLAB development environment.

During execution of the WA, the contractor shall:

- Attempt to hold phone discussions with the WAM approximately biweekly to discuss technical progress on all active requests.
- Notify the EPA WAM via direct email or other automated COLAB email-based communication when a successfully executed test plan (less Agency approval) for a given component has been posted to COLAB.

In closing out given technical direction, the contractor shall provide a Summary Technical Progress Report in email form to the WAM if one or more components were not completed. In this case, the contractor shall briefly summarize (e.g., in simple table format) which deliverables were not completed for each component.

Processing and Documentation of Software Bugs

For each component, until successfully executed test plan (with Agency approval) status has been reached, it is anticipated that the Agency and the contractor may post new bugs that are identified during review and testing associated with a given request. All detailed notations on specific bugs to be resolved and bug resolution will be conducted via COLAB by both EPA and the contractor. Any new bug identified by the contractor during execution of this WA, which substantially changes existing specifications and design shall be posted to COLAB and appropriate notations provided (i.e., the Agency requires that all substantial changes made to the software are documented through COLAB bugs and COLAB notations for component design and specifications). Any bug identified but not resolved by the contractor during execution of this WA, which substantially affects attainment of the component's stated software requirements, shall also be posted to COLAB (i.e., the Agency requires that all known remaining software deficiencies identified by the contractor during testing be documented in COLAB). Minor bugs that are resolved during evaluation, modification, enhancement or testing that do not substantially affect existing design and specifications documentation need not to be notated in COLAB.

Total Task Level of Effort

For purposes of estimating resources for this task the contractor shall assume an overall level of effort of approximately 6,921 hours total of software development, software testing, and project management which will be split across the two tasks and associated Technical Direction.

Deliverables and Schedule:

Because of the nature of the work to be performed, no initial deliverable dates can be set. Specific CO2, SuperMUSE V1, FRAMES V2, and 3MRA V1/V2 software components to be worked on by the contractor, and associated schedule, will be determined during execution of the WA by agreement between the WAM and contractor. In evaluating content and acceptance criteria for deliverables, the following will generally apply:

A. Successfully executed test plan status (less Agency approval) for a given software component requires that:
   1. Specific requirements related to the functionality of the software must be documented (as provided by the WAM within tTechnical Direction);
   2. All identified software bugs have been resolved by the contractor or reconciled as future work to be completed by the Agency (e.g., some bugs may not be able to be resolved at this time within current resources);
   3. Summary notations on modifications and additions to design and specifications sections of formal documentation have been posted to COLAB by the contractor;
   4. Executed and notated test plans have been posted to COLAB by the contractor which satisfy all component requirements; and
   5. Source code and compiled software codes have been posted to COLAB.

B. Successfully executed test plan status (with Agency approval) for a given software component requires that:

   1. Successfully executed test plan status (less Agency approval) has been attained by the contractor for the given software component;

   2. The Agency has reviewed and approved the executed test plan (via email notification to the contactor).

C. Completion Status for Specific Technical Direction

Specific Technical Direction will be deemed complete and no additional efforts should be expended by the contractor on the given request when either:

1. Currently approved hours associated with given Technical Direction have been expended by the contractor and the contractor has provided a Summary Technical Progress Report for all components not completed,

   1.a. Based upon the WAM's assessment of degree of completion, the WAM may reauthorize the existing technical direction by adding additional hours to further complete the specific request. Alternatively, the WAM may also choose to not expend additional effort.

   1.b. In the event that the existing technical direction is re-authorized with additional hours and associated level of effort, the WAM will notify the contractor and EPA's Project Officer by re-issuing and notating the original technical direction, indicating both the previous authorized level already expended, the additional level (i.e., added hours) of effort that may be expended by the contractor, and priorities for the additional level of effort.

   or

2. Successfully executed test plan status (with Agency approval) has been attained for all components identified in the request.

Special Conditions

1. All requests related to execution of the technical support described within this WA shall be coordinated through the EPA WAM.
2. The contractor shall not respond to requests or inquiries made by other individuals except where made by technical support staff approved by the WAM. Approvals will be issued by the WAM by technical direction.
3. It is the responsibility of the contractor to ensure that a Summary Technical Progress Report for all components can be completed for a given request and delivered to the WAM prior to expending all hours for a given request (i.e., as necessary, final hours available for a given request should be used for this tasking).